

WHAT IS CLAIMED IS:

1. A method for dynamic distributed data caching comprising:

5 providing a cache community comprising at least one peer, each peer having an associated first content portion indicating content to be cached by the respective peer;

allowing a client to join the cache community;

10 updating a peer list associated with the cache community to include the client, the peer list indicating the peers in the cache community; and

associating a respective second content portion with each peer based on the addition of the client, the second content portion being distinct from the first content portion.

15 2. The method for dynamic distributed data caching according to Claim 1 and further comprising:

receiving a join request from the client; and

20 determining whether to allow the client to join the cache community.

3. The method dynamic distributed data caching according to Claim 2, wherein the join request comprises a CRMSG_REQUESTTOJOIN data message.

25 4. The method for dynamic distributed data caching according to Claim 1, wherein allowing the client to join the cache community comprises:

generating an allow message;

associating the peer list with the allow message; and

30 communicating the allow message to the client.

5. The method for dynamic distributed data caching according to Claim 4, wherein allowing the client to join the cache community comprises:

generating an allow message comprising the peer list
5 updated to include the client;

communicating the allow message to the client; and

communicating the allow message to at least one member associated with the cache community.

10 6. The method for dynamic distributed data caching according to Claim 4, wherein the allow message comprises a CRMSG_UPDATEPEERLIST data message.

15 7. The method for dynamic distributed caching according to Claim 4, wherein the peer list associated with the allow message comprises the updated peer list which includes the client.

20 8. The method for dynamic distributed data caching according to Claim 1, wherein the peer comprises a computer.

9. The method for dynamic distributed data caching according to Claim 1, wherein a one of the peers comprises a member.

25

10. The method for dynamic distributed data caching according to Claim 1, wherein a one of the peers comprises a master.

11. The method for dynamic distributed data caching according to Claim 1, wherein associating a respective second content portion comprises:

5 allocating respective second content portions to peers in the peer list; and

updating an allocation table to indicate the second content portion associated with the peers.

10 12. The method for dynamic distributed data caching according to Claim 11, wherein the second content portions are 10 distinct.

15 13. The method for dynamic distributed data caching according to Claim 11, wherein at least two of the second content portions overlap.

20 14. The method for dynamic distributed data caching according to Claim 11, wherein the first and second content portions respectively comprise a plurality of internet protocol domain names.

15. The method for dynamic distributed data caching according to Claim 1 and further comprising removing the association between the first content portions and the peers.

16. A system for dynamic distributed data caching comprising:

logic encoded on storage and operable to:

provide a cache community comprising at least one peer, each peer having an associated first content portion indicating content to be cached by the respective peer;

allow a client to join the cache community;

update a peer list associated with the cache community to include the client, the peer list indicating the peers in the cache community; and

associate a respective second content portion with each peer based on the addition of the client, the second content portion being distinct from the first content portion.

15

17. The system for dynamic distributed data caching according to Claim 16, wherein the logic is further operable to:

receive a join request from the client; and

20 determine whether to allow the client to join the cache community.

18. The system for dynamic distributed data caching according to Claim 17, wherein the join request comprises a 25 CRMSG_REQUESTTOJOIN data message.

19. The system for dynamic distributed data caching according to Claim 16, wherein the logic is further operable to:

30 generate an allow message;

associate the peer list with the allow message;

communicate the allow message to the client.

20. The system for dynamic distributed data caching according to Claim 19, wherein the logic is further operable to:

5 generate an allow message comprising the peer list updated to include the client;
communicate the allow message to the client; and
communicate the allow message to at least one member associated with the cache community.

10

21. The system for dynamic distributed data caching according to Claim 19, wherein the allow message comprises a CRMSG_UPDATEPEERLIST data message.

15

22. The system for dynamic distributed caching according to Claim 19, wherein the peer list associated with the allow message comprises the updated peer list which includes the client.

20

23. The system for dynamic distributed data caching according to Claim 16, wherein the peer comprises a computer.

25

24. The system for dynamic distributed data caching according to Claim 16, wherein a one of the peers comprises a member.

25. The system for dynamic distributed data caching according to Claim 16, wherein a one of the peers comprises a master.

30

26. The system for dynamic distributed data caching according to Claim 16, wherein the logic is further operable to:

allocate respective second content portions to peers in
5 the peer list; and

update an allocation table to indicate the second content portion associated with the peers.

27. The system for dynamic distributed data caching
10 according to Claim 26, wherein the second content portions are distinct.

28. The system for dynamic distributed data caching according to Claim 26, wherein at least two of the second
15 content portions overlap.

29. The system for dynamic distributed data caching according to Claim 26, wherein the first and second content portions respectively comprise a plurality of internet
20 protocol domain names.

29. The system for dynamic distributed data caching according to Claim 16, wherein the logic is further operable to remove the association between the first content portions
25 and the peers.

31. A method for dynamic distributed data caching comprising:

determining that a first master associated with a cache community is non-operational;

5 electing a second master to replace the first master in the cache community; and

allocating at least one content portion based on the loss of the first master.

10 32. The method for dynamic distributed data caching according to Claim 31, wherein determining that the first master is non-operational comprises:

determining the amount of time since a member status request message has been received;

15 determining whether the amount of time exceeds a threshold.

33. The method for dynamic distributed data caching according to Claim 32, wherein the member status request message comprises CRMSG_PEERPING data message.

20 34. The method for dynamic distributed data caching according to Claim 32, wherein the threshold comprises a predetermined time.

35. The method for dynamic distributed data caching according to Claim 31, wherein electing the second master comprises:

5 communicating a hold election message to peers associated with the cache community;

receiving at least one response to the hold election message;

selecting a one of the peers to be the second master;

communicating an update master message to a cache server.

10

36. The method for dynamic distributed data caching according to Claim 35, wherein the hold election message comprises a CRMSG_HOLDELECTION data message.

15

37. The method for dynamic distributed data caching according to Claim 35, wherein electing the second master further comprises aborting the election if a response is received from the first master.

20

38. The method for dynamic distributed data caching according to Claim 35, wherein the update master message comprises a CRMSG_REPLACEMASTER data message.

25

39. The method for dynamic distributed data caching according to Claim 31, wherein allocating at least one content portion comprises:

allocating respective content portions to peers in the cache community;

30

updating an allocation table at the second master to indicate the respective content portions associated with the respective peers.

40. The method for dynamic distributed caching according to Claim 39, wherein the content portions are distinct.

41. A system for dynamic distributed data caching comprising:

logic encoded on storage and operable to:

5 determine that a first master associated with a cache community is non-operational;

elect a second master to replace the first master in the cache community; and

allocate at least one content portion based on the loss of the first master.

10

42. The system for dynamic distributed data caching according to Claim 41, wherein the logic is further operable to:

15 determining the amount of time since a member status request message has been received; and

determining whether the amount of time exceeds a threshold.

43. The system for dynamic distributed data caching according to Claim 42, wherein the member status request message comprises CRMSG_PEERPING data message.

44. The system for dynamic distributed data caching according to Claim 45, wherein the threshold comprises a 25 predetermined time.

45. The system for dynamic distributed data caching according to Claim 41, wherein the logic is further operable to:

communicate a hold election message to peers associated
5 with the cache community;

receive at least one response to the hold election message;

select a one of the peers to be the second master; and
communicate an update master message to a cache server.

10

46. The system for dynamic distributed data caching according to Claim 45, wherein the hold election message comprises a CRMSG_HOLDELECTION data message.

15 47. The system for dynamic distributed data caching according to Claim 45, wherein the logic is further operable to abort the election if a response is received from the first master.

20

48. The system for dynamic distributed data caching according to Claim 45, wherein the update master message comprises a CRMSG_REPLACEMASTER data message.

25

49. The system for dynamic distributed data caching according to Claim 41, wherein the logic is further operable to:

allocate respective content portions to peers in the cache community; and

30

update an allocation table at the second master to indicate the respective content portions associated with the respective peers.

50. The system for dynamic distributed caching according to Claim 49, wherein the content portions are distinct.

2010 RELEASE UNDER E.O. 14176

51. A method for dynamic distributed caching comprising:
requesting a list of cache communities from a cache
server;

5 determining whether at least one existing cache community
exists;

attempting to join a one of the existing cache
communities when the existing cache communities are found; and
generating a new cache community when no existing cache
communities are found.

10

52. The method for dynamic distributed caching according
to Claim 51, wherein requesting the list comprises:

generating a community request; and
communicating the community request to the cache server.

15

53. The method for dynamic distributed caching according
to Claim 52 and further comprising determining whether a
response has been received from the cache server within a
threshold time.

20

54. The method for dynamic distributed caching according
to Claim 51, wherein determining whether at least one existing
cache community exists comprises:

25 generating a probe message; and
broadcasting the probe message.

55. The method for dynamic distributed data caching
according to Claim 52, wherein the probe message comprises a
CRMSG_WAKEUP data message.

30

56. The method for dynamic distributed data caching according to Claim 54, wherein the broadcast of the probe message is performed on a internet protocol broadcast address.

5 57. The method for dynamic distributed data caching according to Claim 54 and further comprising determining whether a response has been received from at least one existing cache community within a threshold time.

10 58. The method for dynamic distributed data caching according to Claim 54, wherein broadcasting the probe message comprises multicasting the probe message on an internet protocol multicast address.

15 59. The method for dynamic distributed data caching according to Claim 51, wherein attempting to join the one of the existing cache communities comprises:

generating a join request;

communicating the join request to the one of the existing

20 cache communities.

60. The method for dynamic distributed data caching according to Claim 51, wherein generating the new cache community comprises:

25 generating an add master request; and

communicating the add master request to the cache server.

61. The method for dynamic distributed data caching according to Claim 60, wherein the add master request 30 comprises a CRMSG_ADDMASTER data message.

62. The method for dynamic distributed data caching according to Claim 60 and further comprising listening for a join request.

DIGITAL EVIDENCE
066241.0117

63. A system for dynamic distributed caching comprising:
logic encoded on storage and operable to:

request a list of cache communities from a cache
server;

5 determine whether at least one existing cache
community exists;

attempt to join a one of the existing cache
communities when the existing cache communities are found; and

10 generate a new cache community when no existing
cache communities are found.

64. The system for dynamic distributed caching according
to Claim 63, wherein the logic is further operable to:

generate a community request; and

communicate the community request to the cache server.

65. The system for dynamic distributed caching according
to Claim 64, wherein the logic is further operable to
determine whether a response has been received from the cache
server within a threshold time.

66. The system for dynamic distributed caching according
to Claim 63, wherein the logic is further operable to:

generate a probe message; and

25 broadcast the probe message.

67. The system for dynamic distributed data caching
according to Claim 66, wherein the probe message comprises a
CRMSG_WAKEUP data message.

68. The system for dynamic distributed data caching according to Claim 66, wherein the broadcast of the probe message is performed on an Internet protocol broadcast address.

5

69. The system for dynamic distributed data caching according to Claim 66, wherein the logic is further operable to determine whether a response has been received from at least one existing cache community within a threshold time.

10

70. The system for dynamic distributed data caching according to Claim 66, wherein the logic is further operable to multicast the probe message on an internet protocol multicast address.

15

71. The system for dynamic distributed data caching according to Claim 63, wherein the logic is further operable to:

generate a join request; and

20 communicate the join request to the one of the existing cache communities.

25

72. The system for dynamic distributed data caching according to Claim 63, wherein the logic is further operable to:

generating an add master request; and

communicating the add master request to the cache server.

30

73. The system for dynamic distributed data caching according to Claim 62, wherein the add master request comprises a CRMSG_ADDMASTER data message.

74. The system for dynamic distributed data caching according to Claim 72, wherein the logic is further operable to listen for a join request.

75. A method for dynamic distributed data caching comprising:

generating a content request for requested content at a first peer in a cache community;

5 determining a second peer associated with the requested content, the second peer being associated with the cache community; and

retrieving, by the first peer, the requested content from the second peer.

10

76. The method for dynamic distributed data caching according to Claim 75 and further comprising:

retrieving, by the second peer, the requested content from an origin server when the requested content is unavailable at the second peer; and

15 storing the requested content at the second peer.

77. The method for dynamic distributed data caching according to Claim 76, wherein the requested content is unavailable when the requested content has been expired.

20
25 78. The method for dynamic distributed data caching according to Claim 76, wherein the requested content is unavailable when the requested content is not stored at the second peer.

79. The method for dynamic distributed data caching according to Claim 75, wherein determining the location comprises:

generating, by a cache portion associated with the first

5 peer, a location request;

communicating the location request to a master associated with the cache community; and

receiving a location response from the master, the location response indicating the second peer.

10

80. The method for dynamic distributed data caching according to Claim 79, wherein the cache portion comprises a software application.

15 81. The method for dynamic distributed data caching according to Claim 79 and further comprising forwarding the content request to the second peer.

20 82. The method for dynamic distributed data caching according to Claim 75, wherein the content request comprises a hypertext transport protocol request.

25 83. The method for dynamic distributed data caching according to Claim 75, wherein the content comprises a web page.

84. The method for dynamic distributed data caching according to Claim 75, wherein the first and second peers respectively comprises a member of the cache community.

85. A system for dynamic distributed data caching comprising:

logic encoded on storage and operable to:

5 generate a content request for requested content at a first peer in a cache community;

determine a second peer associated with the requested content, the second peer being associated with the cache community; and

10 retrieve, by the first peer, the requested content from the second peer.

86. The system for dynamic distributed data caching according to Claim 85, wherein the logic is operable to:

15 retrieve, by the second peer, the requested content from an origin server when the requested content is unavailable at the second peer; and

store the requested content at the second peer.

87. The system for dynamic distributed data caching according to Claim 86, wherein the requested content is 20 unavailable when the requested content has been expired.

88. The system for dynamic distributed data caching according to Claim 86, wherein the requested content is 25 unavailable when the requested content is not stored at the second peer.

89. The system for dynamic distributed data caching according to Claim 85, wherein the logic is operable to:

generate, by a cache portion associated with the first peer, a location request;

5 communicate the location request to a master associated with the cache community; and

receive a location response from the master, the location response indicating the second peer.

10 90. The system for dynamic distributed data caching according to Claim 89, wherein the cache portion comprises a software application.

15 91. The system for dynamic distributed data caching according to Claim 89, wherein the logic is operable to forward the content request to the second peer.

20 92. The system for dynamic distributed data caching according to Claim 85, wherein the content request comprises a hypertext transport protocol request.

93. The system for dynamic distributed data caching according to Claim 85, wherein the content comprises a web page.

25

94. The system for dynamic distributed data caching according to Claim 85, wherein the first and second peers respectively comprise a member of the cache community.

95. A method for dynamic distributed data caching comprising:

communicating a community request from a dynamic cache module to an administration module;

5 receiving a community list from the administration module in response to the community request, the community list including a list of communities;

generating a join request to attempt to join a one of the communities in the community list;

10 receiving an allow message associated with the one of the communities;

receiving a peer list associated with the one of the communities;

receiving a content request; and

storing content associated with the content request.

15 96. The method for dynamic distributed data caching according to Claim 95, wherein the community request comprises a CRMSG_WAKEUP data message.

20 97. The method for dynamic distributed data caching according to Claim 95, wherein the join request comprises a CRMSG_REQUESTTOJOIN data message.

98. A system for dynamic distributed data caching comprising:

logic encoded on storage and operable to:

5 communicate a community request from a dynamic cache module to an administration module;

receive a community list from the administration module in response to the community request, the community list including a list of communities;

10 generate a join request to attempt to join a one of the communities in the community list;

receive an allow message associated with the one of the communities;

receive a peer list associated with the one of the communities;

15 receive a content request; and

store content associated with the content request.

15
20 99. The system for dynamic distributed data caching according to Claim 98, wherein the community request comprises a CRMSG_WAKEUP data message.

100. The system for dynamic distributed data caching according to Claim 98, wherein the join request comprises a CRMSG_REQUESTTOJOIN data message.

101. A system for dynamic distributed data caching comprising:

means for providing a cache community comprising at least one peer, each peer having an associated first content portion 5 indicating content to be cached by the respective peer;

means for allowing a client to join the cache community;

means for updating a peer list associated with the cache community to include the client, the peer list indicating the peers in the cache community; and

10 means for associating a respective second content portion with each peer based on the addition of the client, the second content portion being distinct from the first content portion.

2010-04-26 10:26:00

102. A system for dynamic distributed data caching comprising:

means for determining that a first master associated with a cache community is non-operational;

5 means for electing a second master to replace the first master in the cache community; and

means for allocating at least one content portion based on the loss of the first master.

103. A system for dynamic distributed caching comprising:
means for requesting a list of cache communities from a
cache server;

means for determining whether at least one existing cache
5 community exists;

means for attempting to join a one of the existing cache
communities when the existing cache communities are found; and

means for generating a new cache community when no
existing cache communities are found.

104. A system for dynamic distributed data caching comprising:

means for generating a content request for requested content at a first peer in a cache community;

5 means for determining a second peer associated with the requested content, the second peer being associated with the cache community; and

means for retrieving, by the first peer, the requested content from the second peer.

105. A system for dynamic distributed data caching comprising:

means for communicating a community request from a dynamic cache module to an administration module;

5 means for receiving a community list from the administration module in response to the community request, the community list including a list of communities;

means for generating a join request to attempt to join a one of the communities in the community list;

10 means for receiving an allow message associated with the one of the communities;

means for receiving a peer list associated with the one of the communities;

means for receiving a content request; and

15 means for storing content associated with the content request.